 EL978317379 45

ACCESSING DATA PROCESSING SYSTEMS BEHIND A NAT ENABLED NETWORK

BACKGROUND OF THE INVENTION

1. Technical Field:

[0001] The present invention relates in general to improved networking and in particular to a method for accessing data processing systems behind a NAT enabled network. Still more particularly, the present invention relates to receiving a source routing address with a DNS query response, such that loose source routing is enabled for accessing data processing systems behind a NAT enabled network from a client system located outside said NAT enabled network.

2. Description of the Related Art:

[0002] The development of computerized information resources, such as interconnection of computer networks, allows users of data processing systems to link with servers within a network to access vast amounts of electronic information. Multiple types of computer networks have been developed that provide different types of security and access and

operate at different speeds. For example, the internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from the sending network to the protocols used by the receiving network. When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

[0003] For a computer to communicate with other computers and servers on the Internet, it must have an Internet Protocol (IP) address identifying the location of the computer on the network. Thus, an issue facing the Internet is the depletion of address and scaling in routing that arises with the increase in home and business networks.

[0004] Many computers are arranged in a local area network (LAN) or wide area network (WAN) that is a private network used by an individual or business. Computers operating within the private network often have access to the Internet. Thus, an issue facing many individuals and businesses is how to protect data within a local network of computer systems that also have access to the Internet.

[0005] Both the addressing and security problems are often solved using a Network Address Translation (NAT) enabled router with a firewall. When NAT is implemented, the individual machines within a private network have unique private addresses rather than unique public IP address. Thus, a single IP address is used by the NAT router and a port mapping scheme is implemented to route packets to data processing systems in the NAT network. As a result, it is simple for a data processing system in the NAT network to contact an outside system via the Internet because all communications take place using the NAT router's IP address.

Adding additional security, communications routed to the Internet hide the unique local address of the data processing system in the NAT network. Additionally, a port mapping scheme of the NAT router is implanted to route received packets to specific data processing systems in the NAT network.

[0006] While NAT provides solutions to the addressing and protection problems, there are also several disadvantages to NAT. Primarily, while it is easy for machines within the NAT network to contact machines outside the NAT network, the reverse is not true. A user at work may want to access his home machine to download images from the home machine or telnet to the home machine, for example. Current NAT techniques do not allow such access directly to machines within the NAT network. Therefore, it would be advantageous to provide a method, system, and program for accessing data processing systems behind a NAT enabled network. Further, it would be advantageous to provide a method, system, and program for accessing data processing systems behind a NAT enabled network without requiring use of a dedicated port.

SUMMARY OF THE INVENTION

[0007] In view of the foregoing, it is therefore an object of the present invention to provide improved network systems.

[0008] It is another object of the present invention to provide a method, system and program for accessing data processing systems behind a NAT enabled network.

[0009] It is yet another object of the present invention to provide a method, system and program for receiving a source routing address with a DNS query response, such that loose source routing is enabled for accessing data processing systems behind a NAT enabled network from a client system located outside said NAT enabled network.

[0010] According to one aspect of the present invention, a NAT data processing system is located behind a NAT enabled network with a NAT device as a gateway to the NAT enabled network. A client system located outside the NAT enabled network queries the NAT device for the address of the NAT data processing system located behind the NAT enabled network. The query is automatically routed through the NAT device to a DNS server. The DNS server then returns an address for the NAT data processing system and source routing for the NAT device. The NAT device forwards the address and source routing to the client system. Then, the client system sends packets to the NAT data processing system at the address with source routing

through the NAT device, such that the NAT data processing system behind the NAT enabled network is directly accessed by the client system from outside the NAT enabled network.

[0011] In querying the NAT device for the address of the NAT data processing system, the client system first receives a user request to establish a connection with a particular domain name, wherein the domain name identifies the NAT data processing system. The client system then sends a DNS query of the domain name to the NAT device. In particular, the client system may first query a local DNS server with the domain name of the NAT data processing system. If the local DNS server cannot authoritatively return an address for the domain name, then a resolv.conf file is consulted for another address to try the DNS query. Advantageously, the address of the NAT device is designated in the resolv.conf file, so that when the DNS query is sent to the NAT device address, the DNS query is then automatically routed to a DNS server that stores the private address of the NAT data processing system and the source routing for the NAT device.

[0012] Multiple data processing systems may be located behind a NAT enabled network that are parallel in the services and data provided. Thus, when a query is sent to the NAT device to resolve the domain name of the NAT data processing system, the DNS query routed through the NAT device may return the addresses of other parallel data processing systems operating behind the NAT enabled network. Thus, if one of the multiple parallel data processing systems is unavailable, the next one can be tried using the returned address of the next

parallel data processing system and the source routing for the NAT device.

[0013] All objects, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

[0015] **Figure 1** is a block diagram depicting a computer system in which the present method, system, and program may be implemented;

[0016] **Figure 2** is a block diagram depicting a distributed network system for facilitating communications between systems in a NAT network and systems in a public network in accordance with the method, system, and program of the present invention;

[0017] **Figure 3** is a block diagram depicting a distributed network system for accessing a data processing system behind a NAT enabled network in accordance with the method, system, and program of the present invention;

[0018] **Figure 4** is a illustrative representation of the data accessed and routed to access a data processing system behind a NAT enabled network in accordance with the method, system,

and program of the present invention;

[0019] **Figure 5** is a flow diagram depicting the data routed to access a data processing system behind a NAT enabled network in accordance with the method, system, and program of the present invention;

[0020] **Figures 6A-6B** depict a high level logic flowchart of a process and program for accessing a data processing system behind a NAT enabled network; and

[0021] **Figure 7** depicts a high level logic flowchart of a process and program for locating the NAT gateway to then access a data processing system behind a NAT enabled network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0022] Referring now to the drawings and in particular to **Figure 1**, there is depicted one embodiment of a computer system in which the present method, system, and program may be implemented. The present invention may be executed in a variety of systems, including a variety of computing systems and electronic devices under a number of different operating systems. In general, the present invention is executed in a computer system that performs computing tasks such as manipulating data in storage that is accessible to the computer system. In addition, the computer system includes at least one output device and at least one input device.

[0023] Computer system **10** includes a bus **22** or other communication device for communicating information within computer system **10**, and at least one processing device such as processor **12**, coupled to bus **22** for processing information. Bus **22** preferably includes low-latency and higher latency paths that are connected by bridges and adapters and controlled within computer system **10** by multiple bus controllers. When implemented as a server system, computer system **10** typically includes multiple processors designed to improve network servicing power.

[0024] Processor **12** may be a general-purpose processor such as IBM's PowerPC™ processor that, during normal operation, processes data under the control of operating system and application software accessible from a dynamic storage device such as random access memory (RAM) **14** and a static storage device such as Read Only Memory (ROM) **16**. The operating system preferably provides a graphical user interface (GUI) to the user. In a preferred

embodiment, application software contains machine executable instructions that when executed on processor **12** carry out the operations depicted in the flowcharts of **Figures 6, 7,** and others described herein. Alternatively, the steps of the present invention might be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

[0025] The present invention may be provided as a computer program product, included on a machine-readable medium having stored thereon the machine executable instructions used to program computer system **10** to perform a process according to the present invention. The term "machine-readable medium" as used herein includes any medium that participates in providing instructions to processor **12** or other components of computer system **10** for execution. Such a medium may take many forms including, but not limited to, non-volatile media, volatile media, and transmission media. Common forms of non-volatile media include, for example, a floppy disk, a flexible disk, a hard disk, magnetic tape or any other magnetic medium, a compact disc ROM (CD-ROM) or any other optical medium, punch cards or any other physical medium with patterns of holes, a programmable ROM (PROM), an erasable PROM (EPROM), electrically EPROM (EEPROM), a flash memory, any other memory chip or cartridge, or any other medium from which computer system **10** can read and which is suitable for storing instructions. In the present embodiment, an example of a non-volatile medium is mass storage device **18** which as depicted is an internal component of computer system **10**, but will be understood to also be provided by an external device. Volatile media include dynamic memory such as RAM **14**. Transmission media include coaxial cables, copper wire or fiber optics,

including the wires that comprise bus **22**. Transmission media can also take the form of acoustic or light waves, such as those generated during radio frequency or infrared data communications.

[0026] Moreover, the present invention may be downloaded as a computer program product, wherein the program instructions may be transferred from a remote computer such as a server **40** to requesting computer system **10** by way of data signals embodied in a carrier wave or other propagation medium via a network link **34** (e.g., a modem or network connection) to a communications interface **32** coupled to bus **22**. Communications interface **32** provides a two-way data communications coupling to network link **34** that may be connected, for example, to a local area network (LAN), wide area network (WAN), or as depicted herein, directly to an Internet Service Provider (ISP) **37**. In particular, network link **34** may provide wired and/or wireless network communications to one or more networks.

[0027] ISP **37** in turn provides data communication services through network **39**. Network **39** may refer to the worldwide collection of networks and gateways that use a particular protocol, such as Transmission Control Protocol (TCP) and Internet Protocol (IP), to communicate with one another. ISP **37** and network **39** both use electrical, electromagnetic, or optical signals that carry digital data streams. The signals through the various networks and the signals on network link **34** and through communication interface **32**, which carry the digital data to and from computer system **10**, are exemplary forms of carrier waves transporting the information.

[0028] When implemented as a server system, including an Internet Domain Name System (DNS), computer system **10** typically includes multiple communication interfaces

accessible via multiple peripheral component interconnect (PCI) bus bridges connected to an input/output controller. In this manner, computer system **10** allows connections to multiple network computers.

[0029] Further, multiple peripheral components may be added to computer system **10,** connected to multiple controllers, adapters, and expansion slots coupled to one of the multiple levels of bus **22**. For example, an audio input/output **28** is connectively enabled on bus **22** for controlling audio input through a microphone or other sound or lip motion capturing device and for controlling audio output through a speaker or other audio projection device. A display **24** is also connectively enabled on bus **22** for providing visual, tactile or other graphical representation formats. A keyboard **26** and cursor control device **30**, such as a mouse, trackball, or cursor direction keys, are connectively enabled on bus **22** as interfaces for user inputs to computer system **10**. In alternate embodiments of the present invention, additional input and output peripheral components may be added.

[0030] Those of ordinary skill in the art will appreciate that the hardware depicted in **Figure 1** may vary. Furthermore, those of ordinary skill in the art will appreciate that the depicted example is not meant to imply architectural limitations with respect to the present invention.

[0031] With reference now to **Figure 2**, a block diagram depicts a distributed network system for facilitating communications between systems in a NAT network and systems in a public network in accordance with the method, system, and program of the present invention.

Distributed data processing system **41** is a network of computers in which the present invention may be implemented. Distributed data processing system **41** includes a public network, such as Internet **42**, and a private network, such as NAT network **58**. NAT network **58** may be implemented as a LAN, a WAN, or other private network. Internet **42** and NAT network **58** are the mediums used to provide communications links between various devices and computers connected together within distributed data processing system **41**. Internet **42** and NAT network **58** may include permanent connections such as wire or fiber optics cables, temporary connections made through telephone connections and wireless transmission connections.

[0032] In the depicted example, server **43** and client **45** are connected to Internet **42**. In addition, server **44** and client **46** are connected to NAT network **58**. Clients **44** and **45** may be, for example, personal computers or network computers. For purposes of this application, a network computer is any computer coupled to a network, which receives communicates with another computer coupled to the network.

[0033] The client/server environment of distributed data processing system **41** is implemented within many network architectures. For example, the architecture of the World Wide Web (the Web) follows a traditional client/server model environment. The terms "client" and "server" are used to refer to a computer's general role as a requester of data (the client) or provider of data (the server). In the Web environment, web browsers such as Netscape Navigator™ typically reside on client systems **45** and **46** and render Web documents (pages) served by a web server, such as servers **43** and **44**. Additionally, each of client systems **45** and **46** and servers **43** and **44** may function as both a "client" and a "server" and may be implemented

utilizing a computer system such as computer system **10** of **Figure 1**. In the examples described for the present invention, client systems **45** and **46** are engaged in peer-to-peer network communications and downloading. In alternate embodiments of the invention, a client-server network communication is also desireable.

[0034] NAT can be implemented on multiple devices, such as NAT box **54**. NAT box **54** may include a router, a gateway, a firewall, and any other device that sits between NAT network **58** and Internet **42**. In NAT network **58**, client server **44** and client **46** are assigned private addresses. It is typical for data processing systems operating behind NAT network **58** to be assigned private addresses, that are not necessarily globally unique, starting with a network number 10. NAT box **54** is assigned an IP address that is globally unique.

[0035] When client **46** wants to communicate with a data processing system outside NAT network **58**, such as server **43**, NAT box **54** receives the IP packets and translates the IP source address for client **46** from the private address to the IP address assigned to NAT box **54**. When packets come back from a host via Internet **42**, NAT box **54** translates the destination address to the private address of client **46** and forwards the packet to the host.

[0036] According to an advantage of the present invention, when client **45** wants to communicate directly with server **44** or client **46** within NAT network **58**, loose source routing is implemented by client **45**. Client **45** receives the private address of server **44** or client **46** and a source routing address for NAT box **54**. Client **45** sends packets to server **44** or client **46** at the private address with loose source routing enabled with the source routing address. No additional port mapping configurations are required in NAT box **54** for enabling access to server **44** or

client **46**.

[0037] According to another advantage of the present invention, NAT network **58** may include multiple servers, such as server **44**, which provide the same service in NAT network **58**. In this case, when client **45** requests communication for the service provided by the multiple servers, client **45** receives the private addresses of each of the parallel servers and the source routing address for NAT box **54**. The communication may then be routed by NAT box **54**, via loose source routing, to an available server.

[0038] Referring now to **Figure 3**, there is depicted a block diagram of a distributed network system for accessing a data processing system behind a NAT enabled network in accordance with the method, system, and program of the present invention. In an example of the present invention, distributed network system **65** includes an application **51** running on host client **50** that requests a connection with the domain name for home machine **60**. For example, a user may request to download photos stored on home machine **60** through the application running on host client **50**. Home machine **60** is one of multiple data processing systems running behind a NAT enabled network implemented by NAT box **54** and NAT network **58**.

[0039] NAT box **54**, assigned a single IP address, implements NAT. NAT box **54** includes a NAT gateway which implements a port mapping scheme to route packets to the host machines, such as home machine **60** connected to NAT network **58**. Additionally, NAT box **54** may include a firewall to protect against unauthorized access to home machine **60**.

[0040] For host client **50** to connect directly with home machine **60**, multiple steps are

required. In step (1), application **51** requests communication with home machine **60** by the domain name for home machine **60**. In step (2), a DNS query is made by host client **50** to obtain the IP address for the domain name. After any required resolver access to resolv.conf, the DNS query is preferably routed to NAT box **54**. In step (3), NAT box **54** receives the query and routes it to a particular port to be forwarded to DNS server **56**. In particular, the NAT gateway of NAT box **54** may be set up to forward all queries on a particular port, such as port 53, to DNS server **56**. In step (4), the query is forwarded to DNS server **56** for address (A) and source routing (SR) Internet addresses. In step (5), DNS server **56** finds the record for the DNS query and returns A for home machine **60** tagged with SR for NAT box **54**. In step (6), host client **50** sends packets to home machine **60** using loose source routing through NAT box **54**. However, prior to NAT box **54** allowing access to home machine **60**, an additional step may require authorization of the user requesting access to home machine **60**. A pre-selected list of authorized users is accessible to NAT box **54**. A user at host client **50** may enter a password, voice sample, or other input that enables determination of the identity of the user at host client **50**. If the user at host client 50 matches one of the pre-selected user identities, then the user is authorized to access home machine **60**.

[0041] With reference now to **Figure 4**, there is depicted an illustrative representation of the data accessed and routed to access a data processing system behind a NAT enabled network in accordance with the method, system, and program of the present invention. For purposes of example, IP addresses used to access a data processing system behind a NAT

enabled network are depicted. A DNS query **80** includes a question for the DNS server to answer stated as (1) a fully qualified domain name (FQDN) for the DNS domain name "machine1.mydomain.com"; (2) the query type to find an address (A) resource record; and (3) the Internet (IN) class for the DNS domain name. For a TCP connection, a response to DNS query **80** typically includes the following fields: name, value, type, class, time-to-live (TTL). The name is the domain name. The value is the IP address or other value mapped to the domain name. The type includes how the Value field should be interpreted. For example, Type=A indicates the value is an IP address and Type=SR indicates the value is the source routing address for use with loose source routing. The TTL specifies how long the resource record is valid.

[0042] DNS query **80** is sent to a local DNS server. If the local DNS server does know how to return an authoritative DNS for "mydomain.com", then the NAT box's IP address is added, as depicted, as a nameserver entry in resolv.conf **82**. Resolv.conf **82** is a configuration file for the DNS client routines "resolver" which is part of a library. In this particular resolv.conf file, for the host client domain "austin.ibm.com", DNS queries are first routed to the local DNS server located at IP address is 9.3.149.2. If the local DNS server is unable to return an authorizative DNS, then the DNS query is next tried at the NAT box located at IP address 9.53.16.20.

[0043] When the NAT box receives DNS query **80**, the query is automatically forwarded to a particular DNS server that stores the A and SR information for accessing the home machine located at "machine1.mydomain.com". In particular, DNS record **84** illustrates the A and SR information for "machine1.mydomain.com." The A is the IP address for the home

machine. The SR is the IP address for the NAT box.

[0044] The DNS server returns DNS response **86** with the information included in DNS record **84**. In particular, it is advantageous for the DNS server to return a response with both A and SR address so that loose source routing may be implemented to access the home machine.

[0045] Referring now to **Figure 5**, there is depicted a flow diagram of the data routed to access a data processing system behind a NAT enabled network in accordance with the method, system, and program of the present invention. As illustrated at reference numeral **70**, a client system sends an Address (A) DNS query to a local DNS server located at IP address 9.3.149.2. As depicted at reference numeral 72, in the example, the local DNS server is unable to authoritatively return an address for the DNS query and so returns a fail response. After consulting resolv.conf, as illustrated at reference numeral **74**, the A DNS query is sent to the NAT box located at IP address 9.53.16.20. The NAT box routes the DNS query to a designated DNS query port and forwards the query to the DNS server enabled to access an IP address for the host machine. As depicted at reference numeral **76**, the DNS server responds with the A and SR records. The NAT box forwards the response to the client. The client then sends a packet to the home machine located at IP address 10.0.3.31 with loose source routing enabled. With loose source routing enabled, the NAT box forwards loose source routing packets directly to the home machine. Although not depicted, an additional packet exchange may be required to authenticate the user requesting access to the home machine at multiple points during the process, such as when the DNS query is received at the NAT box or when the packet with loose source routing is

received at the NAT box.

[0046] With reference now to **Figures 6A-6B**, there is depicted a high level logic flowchart of a process and program for accessing a data processing system behind a NAT enabled network. As depicted, the process starts at block **100** and thereafter proceeds to block **102**. Block **102** depicts a determination whether the application has a request to establish a connection to a home machine. In particular, the request is to access the home machine located at a particular domain name. If the application does not have a request, then the process iterates at block **102**. If the application does have a request, then the process passes to block **104**. Block **104** illustrates resolving the host name for the request by sending a DNS query, here to "machine1.mydomain.com", and the process passes to block **105**.

[0047] Block **105** depicts sending the query to the local DNS server for "machine1.mydomain.com". Thereafter, the process passes to process A depicted in **Figure 7**. When the process returns from process A depicted in **Figure 7**, the process passes to block **106**.

[0048] Block **106** depicts routing the DNS request to port 53 (or another port for which the NAT box has been enabled for forwarding). Next, block **108** illustrates forwarding the DNS query to a particular DNS server, and the process passes to block **110**.

[0049] Block **110** depicts receiving a DNS query for "machine1.mydomain.com." Next, block **112** illustrates responding with the A record and the SR record (if available) for "machine1.mydomain.com", and the process passes to block **114**. According to one advantage of the present invention, where a home machine is located behind a NAT enabled network,

accessing both the A record and the SR record in a DNS query of the home machine domain name will facilitate loose source routing from the client.

[0050] Block **114** depicts forwarding the A record and SR record (if available) to the host device, and the process passes to block **116**.

[0051] Block **116** depicts a determination whether the response has an SR record. If the response does not have an SR record, then the process passes to block **118** where the normal code path is followed and the process ends. If the response does have an SR record, then the process passes to block **120**. Block **120** illustrates passing the A and SR records to the application, and the process passes to block **122**.

[0052] Block **122** depicts a determination whether the response has an SR record. If the response does not have an SR record, then the process passes to block **130** which depicts sending the packet to the A address. If the response does have an SR record, then the process passes to block **124**. Block **124** depicts sending the packet with source routing enabled, and the process passes to block **126**. In particular, by sending the packet with source routing enabled, loose source routed packets are transferred, as will be understood by one skilled in the art. Block **126** depicts forwarding the packet through loose source routing to the home machine, and the process ends.

[0053] Referring now to **Figure 7**, there is depicted a high level logic flowchart of a process and program for locating the NAT gateway to then access a data processing system behind a NAT enabled network. As illustrated, a process A is initiated in the process depicted in

Figure 6. First, block **152** depicts attempting to forward the DNS query to the NAT box. Next, block **156** illustrates a determination whether the forwarding attempt was successful. If the attempt was successful, then the process returns to **Figure 6**. If the attempt was not successfully, then the process passes to block **158**. Block **158** depicts returning an indicator that the attempt failed. Next, block **160** depicts selecting the next name server from the resolv.conf file, and the process passes to block **152** where the next attempt to forward the query to the NAT box is made to the address identified as the next nameserver.

[0054] While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.